WHAT IS CLAIMED IS:

1. A method for access control comprising the steps of:

receiving a packet at an element having a buffer;

determining if an average buffer fill of the buffer in which the packet is to be stored is greater than a minimum threshold and less than a maximum threshold, where the average buffer fill is an average fill state of the buffer;

calculating a drop probability associated with the packet which identifies a probability the packet will be dropped from the element;

comparing the drop probably to a random number; and

discarding the packet from the element if the drop probability is greater than the random number.

2. A method as described in Claim 1 wherein the calculating step includes the step of retrieving the drop probability from a probability memory having precomputed drop probabilities.

3. A method as described in Claim 2 wherein the calculating step includes before the retrieving step the step of identifying an address in the probability memory of the drop probability.

4. A method as described in Claim 3 wherein the identifying the address step includes the step of mapping values of variables into at least two regions.

5. A method as described in Claim 4 wherein the identifying the address step includes after the mapping step the step of determining the address from the variables.

6. A method as described in Claim 5 wherein the mapping step includes the step of mapping of variables into at least two regions by comparing them with a programable thresholds.

7. A method as described in Claim 6 wherein the comparing step includes the step of comparing the drop probability to the random number generated by a linear feedback shift register which approximates a random number generator.

8. A method as described in Claim 7 wherein the variables includes count since last drop, packet size, average buffer fill, color and queue, respectively.

9. A method as described in Claim 8 wherein the regions include count_region, pktsize_ region, abf_ region, color and queue derived from the variables count since last drop, packet size, average buffer fill, color and queue, respectively.

10. A method as described in Claim 9 wherein the discarding step includes the step of discarding a packet every 2 clock cycles.

11. A method as described in Claim 10 wherein the element includes a reassembler for sending packets to a network.

12. A method as described in Claim 10 wherein the probability is defined as

$$P_a(abf_{marking}) = \frac{p_b(abf_{marking})}{1 - count * p_b(abf_{marking})}$$

where

$$p_b(abf_{marking}) = \left(\frac{\mathrm{max\_p}_{\mathrm{marking}} \cdot (\mathrm{abf}_{\mathrm{marking}} - \mathrm{minth}_{\mathrm{marking}})}{\mathrm{maxth}_{\mathrm{marking}} - \mathrm{minth}_{\mathrm{marking}}}\right)\left(\frac{pktsize}{\mathrm{maxPacket}}\right) \cdot queue\_weight_{queue}$$

and $max\_p_{marking}$, $maxth_{marking}$, $minth_{marking}$, maxPacket, and $queue\_weight_{queue}$ are constants.

13. A method as described in Claim 12 wherein the

$$abf_{marking,new} = \frac{(k_1 \cdot abf_{marking,old} + k_2 \cdot iqf_{marking})}{X}$$

where k_1 and k_2 are integers whose sum = X.

14. A method as described in Claim 13 wherein $X = 64$.

15. An apparatus for storing a packet comprising:

a buffer in which packets are stored;

a mechanism for determining an average buffer fill of the buffer, where the average buffer fill is an average fill state of the buffer, the determining mechanism is connected to the buffer;

a mechanism for calculating a drop probability associated with the packet which identifies the probability the packet will be dropped from the buffer; and

a mechanism for generating a random number; and

a mechanism for discarding the packet from the elements if the drop probability is greater in than the random number.

16. An apparatus as described in Claim 15 wherein the calculating mechanism includes a probability mechanism having predetermined drop probabilities from which the drop probability is obtained.

17. An apparatus as described in Claim 16 wherein the probability mechanism has a probability memory having addresses and the calculating mechanism includes a mechanism for identifying an address in the probability memory having the drop probability.

18. An apparatus as described in Claim 17 wherein the calculating mechanism includes a mechanism for mapping values of variables into regions.

19. An apparatus as described in Claim 18 wherein the random number generating mechanism includes a linear feedback shift register which approximates a random number generator.

20. An apparatus as described in Claim 19 wherein the probability is defined as

$$P_a(abf_{marking}) = \frac{p_b(abf_{marking})}{1 - count * p_b(abf_{marking})}$$

where

$$p_b(abf_{marking}) = \left(\frac{\max\_p_{marking} \cdot (abf_{marking} - minth_{marking})}{maxth_{marking} - minth_{marking}} \right) \left(\frac{pktsize}{maxPacket} \right) \cdot queue\_weight_{queue}$$

and $max\_p_{marking}$, $maxth_{marking}$, $minth_{marking}$, maxPacket, and $queue\_weight_{queue}$ are constants.

21. An apparatus as described in Claim 20 wherein the

$$abf_{marking,new} = \frac{(k_1 \cdot abf_{marking,old} + k_2 \cdot iqf_{marking})}{64}$$

where k_1 and k_2 are integers whose sum = 64.

22. An apparatus as described in Claim 20 wherein the determining mechanism includes a mechanism for determining the instantaneous queue fill.